



Introduction to AWS

Amazon Web Services offers a broad set of global cloud-based products including compute, storage, databases, analytics, networking, mobile, developer tools, management tools, IoT, security, and enterprise applications: on-demand, available in seconds, with pay-as-you-go pricing. From data warehousing to deployment tools, directories to content delivery, over 140 AWS services are available. New services can be provisioned quickly, without the upfront capital expense. This allows enterprises, start-ups, small and medium-sized businesses, and customers in the public sector to access the building blocks they need to respond quickly to changing business requirements.

In 2006, Amazon Web Services (AWS) began offering IT infrastructure services to businesses as web services—now commonly known as cloud computing. One of the key benefits of cloud computing is the opportunity to replace upfront capital infrastructure expenses with low variable costs that scale with your business. With the cloud, businesses no longer need to plan for and procure servers and other IT infrastructure weeks or months in advance. Instead, they can instantly spin up hundreds or thousands of servers in minutes and deliver results faster.

Cloud Computing

Cloud computing is the on-demand delivery of compute power, database storage, applications, and other IT resources through a cloud services platform via the Internet with pay-as-you-go pricing. Whether you are running applications that share photos to millions of mobile users or you're supporting the critical operations of your business, a cloud services platform provides rapid access to flexible and low-cost IT resources. With cloud computing, you don't need to make large upfront investments in hardware and spend a lot of time on the heavy lifting of managing that hardware. Instead, you can provision exactly the right type and size of computing resources you need to power your newest bright idea or operate your IT department. You can access as many resources as you need, almost instantly, and only pay for what you use.

Cloud computing provides a simple way to access servers, storage, databases and a broad set of application services over the Internet. A cloud services platform such as Amazon Web Services owns and maintains the network-connected hardware required for these application services, while you provision and use what you need via a web application.

Advantages of Cloud Computing

- Trade capital expense for variable expense – Instead of having to invest heavily in data centres and servers before you know how you're going to use them, you can pay only when you consume computing resources, and pay only for how much you consume.
- Benefit from massive economies of scale – By using cloud computing, you can achieve a lower variable cost than you can get on your own. Because usage from hundreds of thousands of customers is aggregated in the cloud, providers such as AWS can achieve higher economies of scale, which translates into lower pay as-you-go prices.
- Stop guessing capacity – Eliminate guessing on your infrastructure capacity needs. When you make a capacity decision prior to deploying an application, you often end up either sitting on expensive idle resources or dealing with limited capacity. With cloud computing, these problems go away. You can access as much or as little capacity as you need, and scale up and down as required with only a few minutes' notice.
- Increase speed and agility – In a cloud computing environment, new IT resources are only a click away, which means that you reduce the time to make those resources available to your developers from weeks to just minutes. This results in a dramatic increase in agility for the organisation, since the cost and time it takes to experiment and develop is significantly lower.
- Stop spending money running and maintaining data centres – Focus on projects that differentiate your business, not the infrastructure. Cloud computing lets you focus on your own customers, rather than on the heavy lifting of racking, stacking, and powering servers.
- Go global in minutes – Easily deploy your application in multiple regions around the world with just a few clicks. This means you can provide lower latency and a better experience for your customers at minimal cost.

Types of Cloud Computing

Cloud computing provides developers and IT departments with the ability to focus on what matters most and avoid undifferentiated work such as procurement, maintenance, and capacity planning. As cloud computing has grown in popularity, several different models and deployment strategies have emerged to help meet specific needs of different users. Each type of cloud service and deployment method provides you with different levels of control, flexibility, and management. Understanding the differences between Infrastructure as a Service, Platform as a Service, and Software as a Service, as well as what deployment strategies you can use, can help you decide what set of services is right for your needs.

Cloud Computing Models

- **Infrastructure as a Service (IaaS)**

Infrastructure as a Service (IaaS) contains the basic building blocks for cloud IT and typically provide access to networking features, computers (virtual or on dedicated hardware), and data storage space. IaaS provides you with the highest level of flexibility and management control over your IT resources and is most similar to existing IT resources that many IT departments and developers are familiar with today.

- **Platform as a Service (PaaS)**

Platform as a Service (PaaS) removes the need for your organisation to manage the underlying infrastructure (usually hardware and operating systems) and allows you to focus on the deployment and management of your applications. This helps you be more efficient as you don't need to worry about resource procurement, capacity planning, software maintenance, patching, or any of the other undifferentiated heavy lifting involved in running your application.

- **Software as a Service (SaaS)**

Software as a Service (SaaS) provides you with a completed product that is run and managed by the service provider. In most cases, people referring to Software as a Service are referring to end-user applications. With a SaaS offering you do not have to think about how the service is maintained or how the underlying infrastructure is managed; you only need to think about how you will use that particular piece of software. A common example of a SaaS application is web-based email which you can use to send and receive email without having to manage feature additions to the email product or maintain the servers and operating systems that the email program is running on.



Cloud Computing Deployment Models

- **Cloud**

A cloud-based application is fully deployed in the cloud and all parts of the application run in the cloud. Applications in the cloud have either been created in the cloud or have been migrated from an existing infrastructure to take advantage of the benefits of cloud computing. Cloud-based applications can be built on low-level infrastructure pieces or can use higher level services that provide abstraction from the management, architecting, and scaling requirements of core infrastructure.

- **Hybrid**

A hybrid deployment is a way to connect infrastructure and applications between cloud-based resources and existing resources that are not located in the cloud. The most common method of hybrid deployment is between the cloud and existing on-premises infrastructure to extend, and grow, an organisation's infrastructure into the cloud while connecting cloud resources to the internal system. For more information on how AWS can help you with your hybrid deployment, please visit our hybrid page.

- **On-premises**

The deployment of resources on-premises, using virtualisation and resource management tools, is sometimes called the “private cloud.” On-premises deployment doesn’t provide many of the benefits of cloud computing but is sometimes sought for its ability to provide dedicated resources. In most cases this deployment model is the same as legacy IT infrastructure while using application management and virtualisation technologies to try and increase resource utilisation.

AWS Regions

The AWS Cloud infrastructure is built around AWS Regions and Availability Zones. An AWS Region is a physical location in the world where we have multiple Availability Zones. Availability Zones consist of one or more discrete data centres, each with redundant power, networking, and connectivity, housed in separate facilities. These Availability Zones offer you the ability to operate production applications and databases that are more highly available, fault tolerant, and scalable than would be possible from a single data centre. The AWS Cloud operates in over 60 Availability Zones within over 20 geographic Regions around the world, with announced plans for more Availability Zones and Regions.

Each Amazon Region is designed to be completely isolated from the other Amazon Regions. This achieves the greatest possible fault tolerance and stability. Each Availability Zone is isolated, but the Availability Zones in a Region are connected through low-latency links. AWS provides you with the flexibility to place instances and store data within



multiple geographic regions as well as across multiple Availability Zones within each AWS Region. Each Availability Zone is designed as an independent failure zone.

This means that Availability Zones are physically separated within a typical metropolitan region and are located in lower risk flood plains (specific flood zone categorisation varies by AWS Region). In addition to discrete uninterruptible power supply (UPS) and onsite backup generation facilities, they are each fed via different grids from independent utilities to further reduce single points of failure. Availability Zones are all redundantly connected to multiple tier-1 transit providers.

Security and Compliance

Security

Cloud security at AWS is the highest priority. Security in the cloud is much like security in your on-premises data centres—only without the costs of maintaining facilities and hardware. In the cloud, you don't have to manage physical servers or storage devices. Instead, you use software-based security tools to monitor and protect the flow of information into and of out of your cloud resources.

The AWS Cloud enables a shared responsibility model. While AWS manages security of the cloud, you are responsible for security in the cloud. This means that you retain control of the security you choose to implement to protect your own content, platform, applications, systems, and networks no differently than you would in an on-site data centre.

Finally, AWS environments are continuously audited, with certifications from accreditation bodies across geographies and verticals. In the AWS environment, you can take advantage of automated tools for asset inventory and privileged access reporting.



Benefits of AWS Security

- **Keep Your Data Safe:** The AWS infrastructure puts strong safeguards in place to help protect your privacy. All data is stored in highly secure AWS data centres.
- **Meet Compliance Requirements:** AWS manages dozens of compliance programs in its infrastructure. This means that segments of your compliance have already been completed.
- **Save Money:** Cut costs by using AWS data centres. Maintain the highest standard of security without having to manage your own facility
- **Scale Quickly:** Security scales with your AWS Cloud usage. No matter the size of your business, the AWS infrastructure is designed to keep your data safe.

Compliance

AWS Cloud Compliance enables you to understand the robust controls in place at AWS to maintain security and data protection in the cloud. As systems are built on top of AWS Cloud infrastructure, compliance responsibilities will be shared. By tying together governance-focused, audit-friendly service features with applicable compliance or audit standards, AWS Compliance enablers build on traditional programs. This helps customers to establish and operate in an AWS security control environment.

Identity and Access Management

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. When a principal makes a request in AWS, the AWS enforcement code checks whether the principal is authenticated (signed in) and authorised (has permissions). You manage access in AWS by creating policies and attaching them to IAM identities or AWS resources. Policies are JSON documents in AWS that, when attached to an identity or resource, define their permissions.

IAM Working

IAM provides the infrastructure necessary to control authentication and authorisation for your account. The IAM infrastructure includes the following elements:

• Terms

Resources

The user, group, role, policy, and identity provider objects that are stored in IAM. As with other AWS services, you can add, edit, and remove resources from IAM.

Identities

The IAM resource objects that are used to identify and group. You can attach a policy to an IAM identity. These include users, groups, and roles.

Entities

The IAM resource objects that AWS uses for authentication. These include IAM users, federated users, and assumed IAM roles.

Principals

A person or application that uses the AWS account root user, an IAM user, or an IAM role to sign in and make requests to AWS.

• Principal

A *principal* is a person or application that can make a request for an action or operation on an AWS resource. The principal is authenticated as the AWS account root user or an IAM entity to make requests to AWS.

• Request

When a principal tries to use the AWS Management Console, the AWS API, or the AWS CLI, that principal sends a *request* to AWS. The request includes the following information:



- **Actions or operations** – The actions or operations that the principal wants to perform. This can be an action in the AWS Management Console, or an operation in the AWS CLI or AWS API.
- **Resources** – The AWS resource object upon which the actions or operations are performed.
- **Principal** – The person or application that used an entity (user or role) to send the request. Information about the principal includes the policies that are associated with the entity that the principal used to sign in.
- **Environment data** – Information about the IP address, user agent, SSL enabled status, or the time of day.
- **Resource data** – Data related to the resource that is being requested. This can include information such as a DynamoDB table name or a tag on an Amazon EC2 instance.

AWS gathers the request information into a *request context*, which is used to evaluate and authorise the request.

• Authentication

A principal must be authenticated (signed in to AWS) using their credentials to send a request to AWS. Some services, such as Amazon S3 and AWS STS, allow a few requests from anonymous users. However, they are the exception to the rule.

To authenticate from the console as a root user, you must sign in with your email address and password. As an IAM user, provide your account ID or alias, and then your user name and password. To authenticate from the API or AWS CLI, you must provide your access key and secret key. You might also be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account.

• Authorization

You must also be authorised (allowed) to complete your request. During authorisation, AWS uses values from the request context to check for policies that apply to the request. It then uses the policies to determine whether to allow or deny the request. Most policies are stored in AWS as JSON documents and specify the permissions for principal entities.

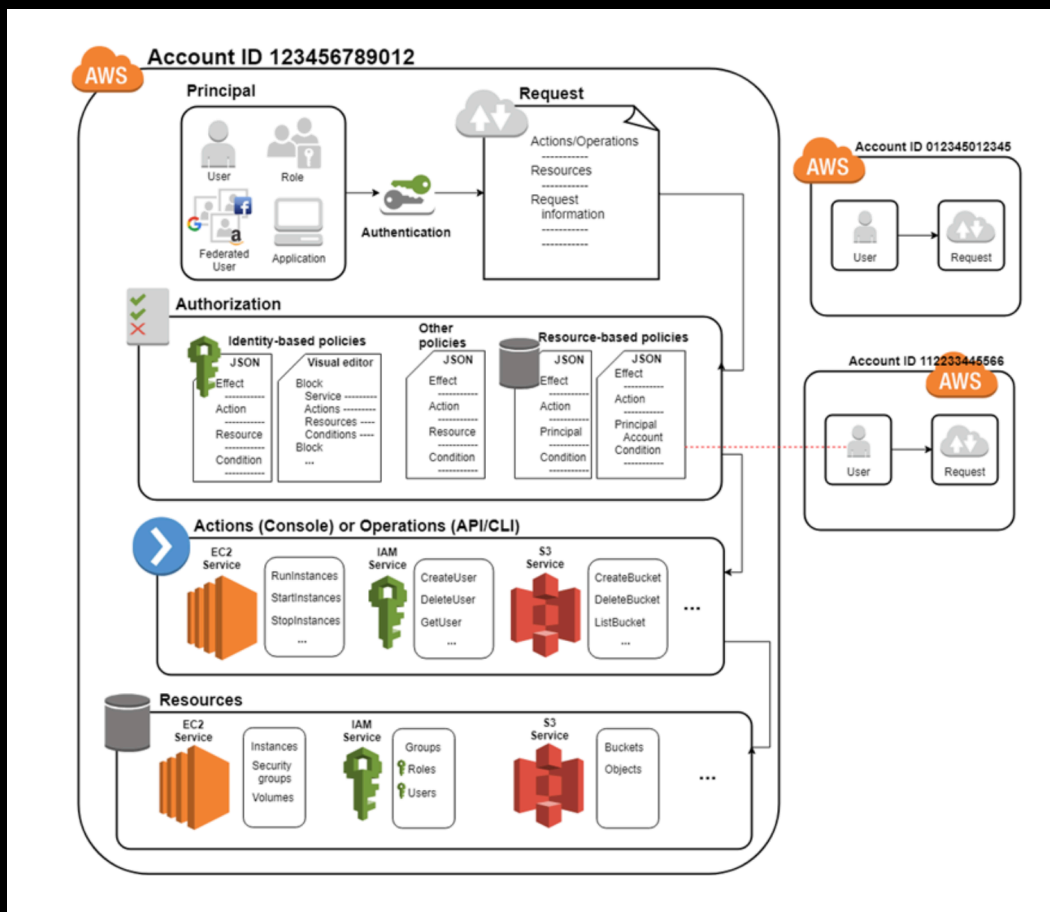
• Actions or Operations

After your request has been authenticated and authorised, AWS approves the actions or operations in your request. Operations are defined by a service, and include things that you can do to a resource, such as viewing, creating, editing, and deleting that resource. For example, IAM supports approximately 40 actions for a user resource, including the following actions:

- CreateUser
- DeleteUser
- GetUser
- UpdateUser

• Resources

After AWS approves the operations in your request, they can be performed on the related resources within your account. A resource is an object that exists within a service. Examples include an Amazon EC2 instance, an IAM user, and an Amazon S3 bucket. The service defines a set of actions that can be performed on each resource.



IAM Users

The "identity" aspect of AWS Identity and Access Management (IAM) helps you with the question "Who is that user?", often referred to as *authentication*. Instead of sharing your root user credentials with others, you can create individual IAM users within your account that correspond to users in your organisation. IAM users are not separate accounts; they are users within your account. Each user can have its own password for access to the AWS Management Console. You can also create an individual access key for each user so that the user can make programmatic requests to work with resources in your account.

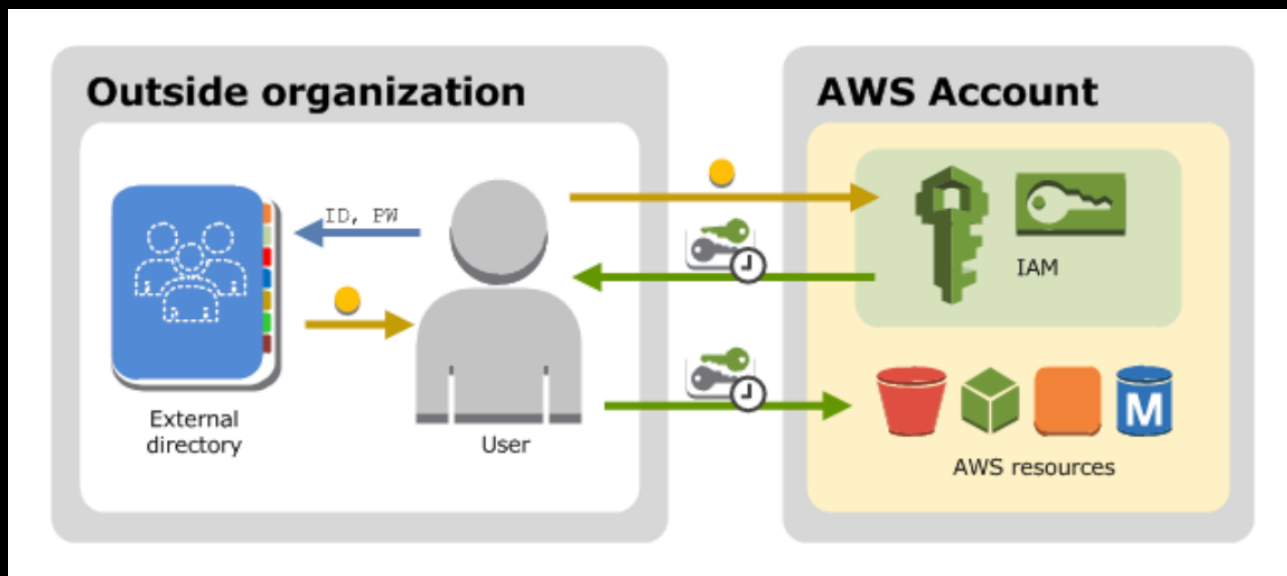


Notice that some of the users are actually applications (for example, DevApp1). An IAM user doesn't have to represent an actual person; you can create an IAM user in order to generate an access key for an application that runs in your corporate network and needs AWS access.

Federating Existing Users

If the users in your organisation already have a way to be authenticated, such as by signing in to your corporate network, you don't have to create separate IAM users for them. Instead, you can *federate* those user identities into AWS.

The following diagram shows how a user can use IAM to get temporary AWS security credentials to access resources in your AWS account.



Using MFA Devices with your IAM Sign-In Page

IAM users who are configured with multi-factor authentication (MFA) devices must use their MFA devices to sign in to the AWS Management Console. After the user enters the user name and password, AWS checks the user's account to see if MFA is required for that user.

If MFA is required for the user, a second sign-in page appears. In the **MFA code** box, the user must enter the numeric code provided by a hardware MFA device.

If the MFA code is correct, the user can access the AWS Management Console. If the code is incorrect, the user can try again with another code.



A hardware MFA device can go out of sync. If a user cannot sign in to the AWS Management Console after several tries, the user is prompted to synchronise the MFA token device. The user can follow the on-screen prompts to synchronise the MFA token device

IAM Roles

An IAM *role* is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session.

You can use roles to delegate access to users, applications, or services that don't normally have access to your AWS resources. For example, you might want to grant users in your AWS account access to resources they don't usually have, or grant users in one AWS account access to resources in another account. Or you might want to allow a mobile app to use AWS resources, but not want to embed AWS keys within the app (where they can be difficult to rotate and where users can potentially extract them). Sometimes you want to give AWS access to users who already have identities defined outside of AWS, such as in your corporate directory. Or, you might want to grant access to your account to third parties so that they can perform an audit on your resources. For these scenarios, you can delegate access to AWS resources using an **IAM role**.

IAM Groups

An IAM group is a collection of IAM users. Groups let you specify permissions for multiple users, which can make it easier to manage the permissions for those users. For example, you could have a group called *Admins* and give that group the types of permissions that administrators typically need. Any user in that group automatically has the permissions that are assigned to the group. If a new user joins your organization and needs administrator privileges, you can assign the appropriate permissions by adding the user to that group. Similarly, if a person changes jobs in your organization, instead of editing that user's permissions, you can remove him or her from the old groups and add him or her to the appropriate new groups.

Note that a group is not truly an "identity" in IAM because it cannot be identified as a **Principal** in a permission policy. It is simply a way to attach policies to multiple users at one time.



Following are some important characteristics of groups:

- A group can contain many users, and a user can belong to multiple groups.
- Groups can't be nested; they can contain only users, not other groups.
- There's no default group that automatically includes all users in the AWS account. If you want to have a group like that, you need to create it and assign each new user to it.
- There's a limit to the number of groups you can have, and a limit to how many groups a user can be in.